**University Of Central Punjab**

****

**Object Oriented Paradigm**

**Lab Manual(Lab 4)**

**Lab Topic:** Constructors and Destructors

**Task 1:**

Suppose we have an online shopping web application. Users are able to choose products from the catalogue. For example, Ali has log in to system and he visited the catalogue page and see the item (Like a shirt, cell phone or a chocolate pack). Ali adds the item to his basket and looks for other items to shop very gently. After the shopping Ali checks out.

You have to create an invoice object that contains all the information necessary to process one line of an invoice. Assume that the invoice must include the following data and functions

**Data:**

• Quantity ordered

• Quantity shipped

• Part number

• Part description

• Unit price

• Extended price

• Sales tax rate

• Sales tax amount

• Shipping

• Total

**Function:**

• A function to initialize all the data items to 0, except the sales tax rate, which is initialized with 5%

• A function to allow the user to initialize all the items by themselves

• A function to calculate the sales tax amount

• A function to calculate the extended cost

• A function to the total amount of the invoice

• A function to display all the information in the invoice

Note: an extended cost is the unit cost multiplied by the number of those items that were purchased. For example, four apples purchased at a unit cost of 1 PKR have an extended cost of 4 PKR (=1 PKR × 4 apples)

**Task 2:**

Write a C++ program that includes the CPU class interfaces, class definitions, and code that will test your classes. Write a main() function along with other functions/procedures, as described below:

The CPU class is described as follows:

* One attribute:
  + clockSpeed – a double value representing the clock speed of the CPU object. Note: clock speed of a modern computer may be something like 2.5GHz, so a value you could use for testing purposes is 2.5.
* Two constructors:
  + A default constructor.
  + A constructor that initializes the attribute, where cs is the clockSpeed.
* Two methods:
  + getClockSpeed() – return the clockSpeed value associated with the CPU object.
  + setClockSpeed(cs) – change the clockSpeed value associated with the CPU object, where cs is the new value.

**Task 3:**

Create a class called Rational for performing arithmetic with fractions. Write a driver program to test your class. Use integer variables to represent the private data of the class, the numerator and the denominator. Provide a constructor function that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initializers are provided and should store the fraction in reduced form (i.e., the fraction 2/4 would be stored in the object as 1 in the numerator and 2 in the denominator). Provide public member functions for each of the following:

1. Addition of two Rational numbers. The result should be stored in reduced form.
2. Subtraction of two Rational numbers. The result should be stored in reduced form.
3. Multiplication of two Rational numbers. The result should be stored in reduced form.
4. Division of two Rational numbers. The result should be stored in reduced form.
5. Printing Rational numbers in the form a/b where a is the numerator and b is the denominator.
6. Printing Rational numbers in double floating-point format.